第 11 部

ポケットベルサービス (WIDE/PCS)

第 1 章

はじめに

広域大規模分散環境の発展は多数の計算機の相互接続を実現し，その結果数多くの利用者が毎日コンピュータネットワークの恩恵を受けるようになった．このような環境下では，ひとたび障害が発生しネットワークサービスが停止すると多くの利用者に多大な影響が及ぶ．このため，ネットワーク管理者は，以前にも増して障害発生時には迅速に対応しなければならない．また，障害に至る前にその可能性を発見し，対策を講じて障害を回避することいった作業も重要度を増している．

このためには，(1) システムを常に適正な状態にあるよう制御する機能，(2) システムの状態を必要なときに必要な形式で報告する機能，(3) 自動的に対処することができないような事態に陥ったときにそれを管理者に通知する機能，をそれぞれ充実させる必要があり，実際にさまざまなアプローチが行なわれている．なかでも SNMP の利用は，近年注目を集めている手法の一つであり，これを利用したアプリケーションも開発されつつあるものの，まだ実験段階であり「決定打を欠い」ているのが現状である．しかし，広域大規模分散環境の急速な拡大を考えると，このような状況からは早急に離脱しなければならないことはあきらかで，そうしなければ今後もさまざまな障害に多くの利用者が悩まされることになる．

このような背景から，以前より筆者はネットワーク管理を支援するシステムについて関心を抱いており，最近では「ネットワーク管理」という作業を図 1.1 のように摸式化し，このモデルに基づいて必要な機能を検討したり実装する作業を行なっている．

このうち，既に発表した `WIDE/PhoneShell` [?] [?] [?] は，上記の (1) や (2) を念頭においたシステムである．そして今回新たに注目したのが，上記の (3) (図 1.1 では，*INTERRUPT* に相当) に適したシステムのあり方で，特に液晶ディスプレイを持つページャ[1]を活用することを検討した．そしてこれを `WIDE/PhoneShell` や， lpr コマンドから利用できる FAX サービス [?] などと併用すると，ネットワーク管理者用メッセージ伝達システムとして有効に利用できると予想した．

そこで本研究では，(1) まずページャをネットワーク環境から制御するシステムを実装し，(2) さらにメッセージ伝達システムとしての利用可能性を検討する，ことを目標とした．

なお，このページャを用いたメッセージ伝達システムを以後 `WIDE/PCS` (WIDE Pager

[1] 日本ではポケットベルあるいはポケベルという表現が普及しているが，英語では pager あるいは beeper という．本論文では pager の日本語読みである「ページャ」という表記を採用した．

Control System) [?] と呼ぶ.

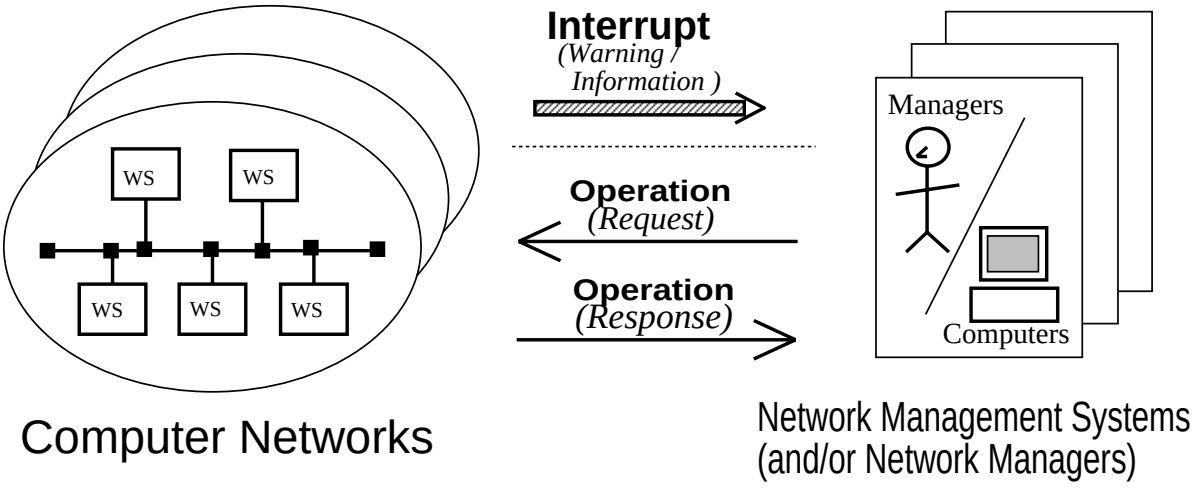


図 1.1: ネットワーク管理のモデル

第 2 章

ページャを用いたメッセージ伝達システム

2.1 ページャの種類

数年前までは，ページャは音を出すことしかできなかったが，現在ではさまざまな製品があり，音だけでなく振動によって着信を知らせる製品や，液晶ディスプレイに文字を表示できる製品などが現れた．このうち，液晶ディスプレイを持つページャは，およそ図 2.1 のように分類できる[1].

分　　類	機　　能	ディスプレイの大きさ
数字型 [2]	12 ないし 14 桁の任意の数字列を表示	12〜14 文字× 1 行
組合せ型 [2]	事前に登録した漢字を含む文字列をいくつか組み合わせ文章にして表示	8 文字× 4 行
カタカナ型 [3]	40 文字まで任意の英数字，カタカナを表示	20 文字× 2 行
自由文型 [4]	36 文字までの漢字を含む任意の文字列を表示	8 文字× 4 行

図 2.1: 液晶ディスプレイを持つページャの種類と機能

ディスプレイを持つページャに文字を表示する方法は，ページャによって少しずつ異るが，基本的には (1) 契約時に指定される局番を呼びだし，(2) 呼が確立したら，機種ごとに決まっている「文字—DTMF 信号対応表」に従って，メッセージを DTMF 信号列として送出する，という手順を踏む．(2) に関して補足すると，例えば NTT の「自由文型」ページャ(以下単に「自由文型」) の場合には，漢字 1 文字を表現するには，まず # を押し # に対応する DTMF 信号を送出し，続いて「表現したい文字の JIS 区点コード」を送出することになっている．すなわち，1 文字を表現するのに DTMF 信号を 5 つ送

[1] これらの製品の中には，現時点ではサービス地域が限定されているものもあるが，数字型，組合せ型，自由文型は，少なくとも東京および大阪では利用可能である．

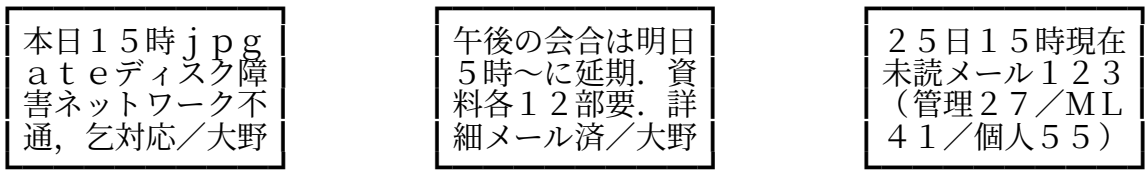
[2] 関東地区では，NTT と 東京テレメッセージ 双方が提供している

[3] 関東地区では，東京テレメッセージのみが提供している

[4] 関東地区では，NTT のみが提供している

出する必要があり,「漢」という文字（JIS 区点コードは 2033）を表示するには，[#] [2] [0] [3] [3]という DTMF 信号を送出しなければならない.

この「自由文型」ページャは，１回の呼びだし操作で最大 36 文字からなるメッセージを表示させることができるので，たとえば図 2.2 のような文章を表示する能力[5] がある．これらの例から，簡単なメッセージが伝達できることがわかるが，そのためには何桁もの DTMF 信号を誤りなく送り出さなければならない．例に示したメッセージの長さはいずれも 32 文字であるが，もし最長の 36 文字からなるメッセージを送る場合には，180 回もの DTMF 信号の送出操作が必要となる．



（Ａ）ネットワークトラブルの報告　　（Ｂ）日程変更の連絡　　（Ｃ）未読メールの蓄積状況の通知

図 2.2: 「自由文型」ページャのメッセージ表示例

このため，人間がプッシュホンを操作して手動でメッセージを送り出すのは現実的な利用法ではなく，何らかの支援環境が必要となる．事実 NTT でもパソコン通信を利用してメッセージを転送するサービスを提供している[6] ．このように,「自由文型」ページャには

- 文字数に制限があり，電子メールのような自由な文章表現はできない.
- 人間がプッシュホンを直接操作して文字を送出するのは現実的ではない.

といった問題があるが,

- 表現を短くする工夫をする.
- 利用支援環境を整備する

ことで，さまざまな情報を伝達するサービスを構築できる.

同様に,「カタカナ型」も利用可能であるが，それ以外の機種では数字とわずかな記号しか表現できないので，あらかじめ「番号とメッセージの対応表」を用意し，利用者は常に対応表を携帯して送られた番号を解読する，といったかなり制限された環境しか提供できない.

[5]「自由文型」の画面のサイズは 8 文字× 4 行（32 文字）である．メッセージが 32 文字を越えた場合には，33 文字目以降は次画面に表示される.

[6]数字だけのメッセージや，あらかじめ決められたメッセージ（定型文）を表示するだけなら，必要な操作は大幅に減少する.

なお，これらのページャはいずれも複数のメッセージ (5 ないし 6 以上) をページャ内部に蓄えることができるので，メッセージを分割して送出すれば，より長い文章を伝達することも不可能ではない．

以上,「自由文型」を中心に液晶ディスプレイをもつページャの特徴の一部を概観した．本論文では，以後「自由文型」に着目し，このページャを計算機から制御する意義，手法およびその応用について議論を進める．

2.2 ページャの問題点と他の手法との比較

前節での検討からも，ページャは「ネットワーク管理者にシステムの異常を報告する装置」(以後，メッセージ伝達装置) としての利用価値があることがわかった．しかし，

1. 伝達が一方通行である．
2. 受信側がサービスエリア外にいた場合，エリア内でも地下などにいた場合には伝達が保証されない．
3. 2 のような事態が生じても送信側はそれを知ることができない

といった欠点がある．メッセージ送信側が伝達を確認するためには，受信側が送信側に対して何らかの応答を行えばよいので，受信者が

- 手近の電話で (音声で) 応答する，
- 送信者もページャを持っている場合には，送信者のページャに確認した旨のメッセージを送る，

などの方法を用いればよい．これらの方法は送信側が人間であれば有効であり，事実これが標準的なページャの利用方法であるが，WIDE/PCS では計算機がページャを駆動し，メッセージを伝達するので，受信側は計算機に対して応答することになり，通常とは別の方法が必要となる．このためには，音声認識装置を用いたり，WIDE/PhoneShell のような DTMF 信号を用いたデータ入力システムを併用するのが妥当だと思われる．さらに，メッセージ送出後一定時間応答がなかった場合には，メッセージの伝達に失敗したとみなし，再送を行なうサービスも必要である．

このように改善の余地はあるものの，計算機が人間にメッセージを伝えるシステムにおいては，メッセージ伝達を保証したければページャだけにたよる利用には無理があり，携帯電話などを利用し，

- 計算機は，音声合成装置を利用し音声によって人間にメッセージを伝える．
- 受信者は，DTMF 信号を用いて計算機に応答する．

といった機構が必要となる．この方法では，電話に応答したのが本人であるのか否かを確認する手段として暗証番号が使えるので，転送の成否が確実に把握できるという利点もあり，ネットワークの稼働状況を監視したり制御するツールとして有効だと思われる．さらに，筆者の周辺では急速に普及しているので，現在積極的に開発を進めている．し

かし，携帯電話はページャに比べると運用経費が高い，体積大きく重い，電池の寿命が短くしばしば充電が必要，電波感度の低下に敏感に反応してしまう，といった欠点もあるため，まずページャを有効に利用するシステムの開発を優先した．

これ以外にも，最近開発された「電子手帳と小型無線データ端末を組み合わせたデータ通信システム」を利用することなども考えられる．この方法は将来的には有望であるが，現時点では仕様等が明確でなく，普及も始まっていない．

2.3　WIDE/PCS の導入により期待される効果

ネットワークからページャを制御するサービスには上記のような問題点があるが，計算機を利用していない，あるいは利用が困難な状況にいるネットワーク管理者に対して，

- システムの異常発生を伝える．
- 他の管理者の連絡先を知らせて互いに連絡をとりあうよう指示する．

といったサービスを提供できる．すなわちこのようなサービスは，広域大規模分散環境におけるネットワーク管理者に新しいメッセージ伝達機構を提供し，重大な障害が生じた場合でも迅速な対応を可能にするので，ネットワークの安定稼働に大きく貢献すると予想される．なお，いかにして異常を自動的に検出するか，あるいは突発した異常に対してどのような仕組みを用意すれば自動的に対応できるかといった問題は，ネットワーク管理に関連したテーマとして興味深いものであり，これらの技術が充実してこそ WIDE/PCS のようなシステムも有効に機能するわけであるが，この問題に関しては今後の課題とし，本論文では議論しない．

第 3 章

試作したシステムの構成と評価

ページャを用いたメッセージ伝達機構の有効性を確認するために評価システムを用意し，1991 年 5 月末から試験運用を開始した．評価システムは「ページャ制御サブシステム」と「フロントエンドサブシステム」からなり，ソフトウェア的には 図 3.1 のような構成になっている．また，ハードウェア的には，WIDE/PhoneShell や FAX サービスなどと機材の一部を共有しているので，図 3.2 のような構成になっている．

現在，この評価システムが提供しているのは，サービスホスト (WIDE/PCS を起動し，サービスを提供しているホスト) 上の疑似ユーザにメッセージを電子メールで送付すると，そのメッセージをページャに転送するというサービスで，1992 年 3 月末現在商品化されているページャには全て対応している．[1]

3.1 ページャ制御サブシステム

ページャ制御サブシステムは，以下に述べる 2 つのサーバ (通常デーモンとして稼働) から構成されている．

一方は *wncud* [2] と呼ばれ，NCU(網制御装置) を制御し，ページャの駆動に必要となる電話回線の制御や DTMF 信号の発信を行なう．もう一方は *pcsd* と呼ばれ，フロントエンドサブシステム (後述) から送られてくる『リクエスト』と呼ばれる「ページャ制御要求」を受け付けて *wncud* に転送したり，*wncud* からの応答をフロントエンドサブシステムに返送する役割を担う．

3.2 wncud

実際に電話回線を制御したり DTMF 信号を発信したりするためには，NCU あるいは NCU 内蔵モデムを用いるのが一般的であるが，これらの装置は通常 RS-232C インタ

[1] もちろん，数字型のページャに漢字のメッセージを送ったような場合には，対応しきれないのでエラーとなる．

[2] 以前は pncd (PNC–3400 control daemon の意) と呼んでおり，そのように表記した論文もあるが，WIDE プロジェクト内で名前の良く似たワーキンググループ (wpnc) が活発な活動を開始したことと，本システムの規模が大きくなるにつれ，pncd という名前が必ずしも実体を的確に表さなくなったことから 1992 年初頭に wncud に名称を変更した．

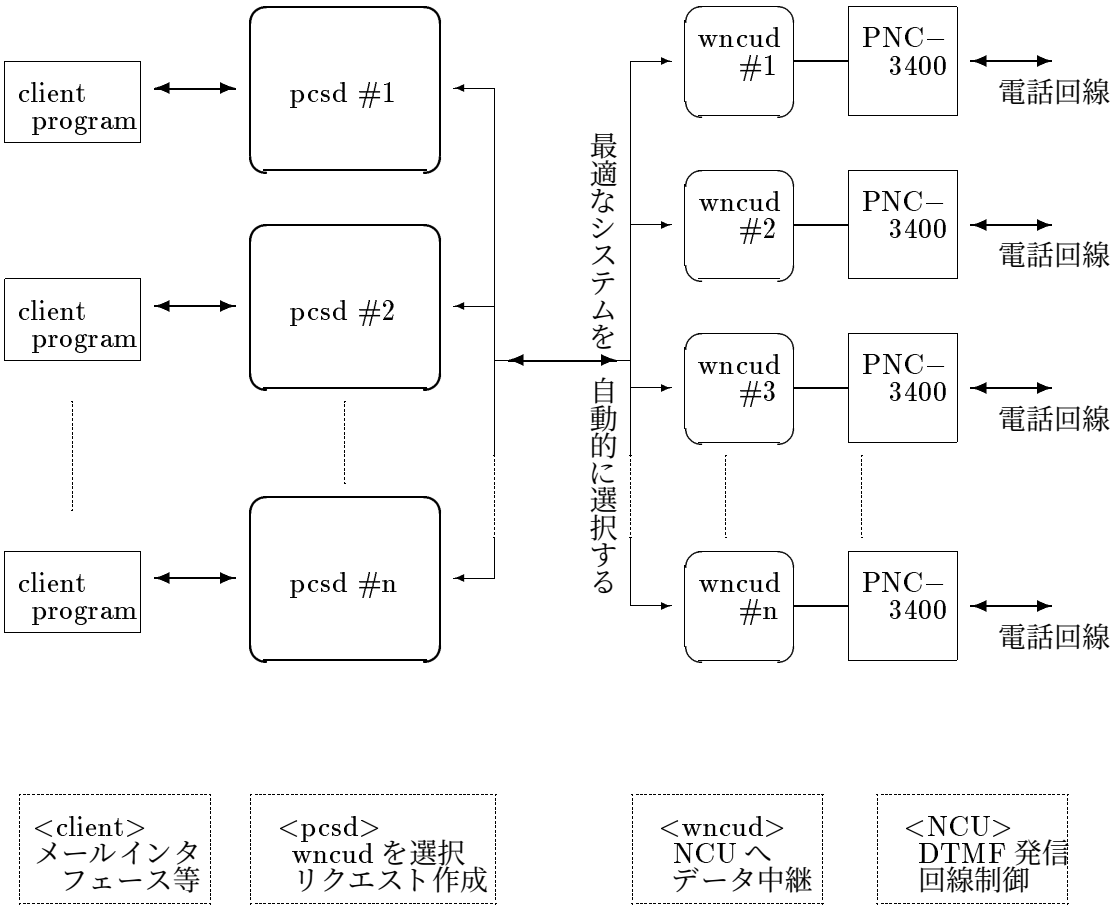


図 3.1: WIDE/PCS 評価システムのソフトウェア構成

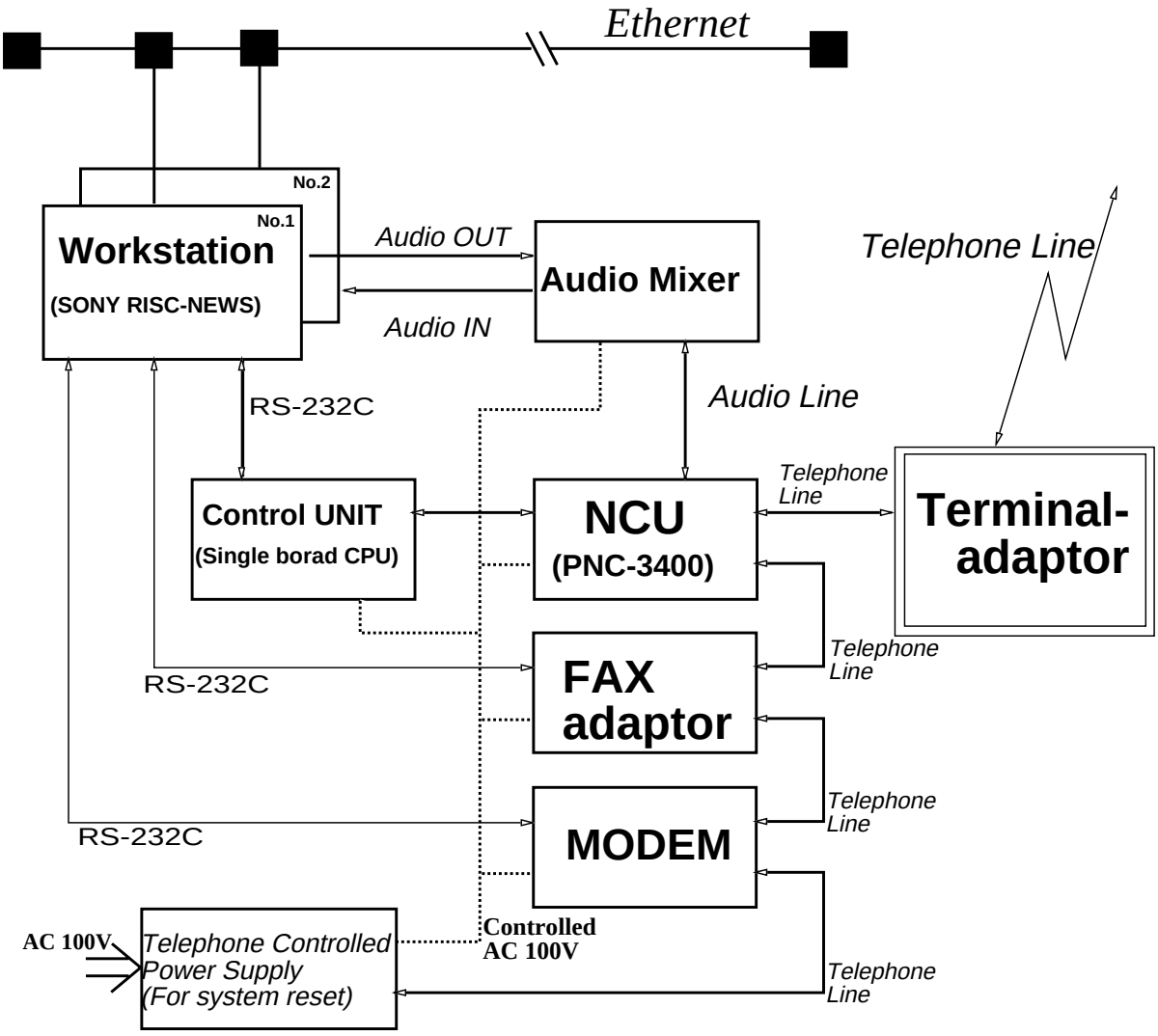


図 3.2: WIDE/PCS 評価システムのハードウェア構成

フェースしか持たないため，ネットワーク経由でこれらの装置を直接制御することはできない．そこで，他のプロセスからデータを受けとり NCU に転送する (あるいはその逆を行なう) 中継プログラムを用意した．この中継プログラムが *wncud* で，起動時の指定により，

- V.25bis をベースとしたコマンド体系を持ち `WIDE/PhoneShell` で採用している PNC–3400
- Hayes 社の AT コマンドをサポートしている通常の NCU 内蔵モデム

のいずれかを制御することができる．*wncud* は原則としてデータを加工せず中継サービスを行なうだけだが，デッドロック防止のためのタイムアウト機能が付加してある．すなわち，NCU が相手局と呼を確立し， *wncud* が NCU にデータを中継している際に，*wncud* にデータを送っていたプロセスがデッドロックあるいは異常終了した場合，何の対策も講じなければ電話回線がオフフックされたままになってしまうことがあり得る．このような問題を回避するため，*wncud* は一定時間データの流れがないと *wncud* のクライアントと NCU の双方の稼働状況をチェックし，必要に応じて回線をオンフックするようにしている．なお *wncud* は，NCU 1 台毎に 1 つずつ起動する必要がある[3]．

3.3　pcsd

pcsd は，次項で述べる「フロントエンドサブシステム」からの『リクエスト』を受け付け，*wncud* に中継する働きをする．多くの場合 *wncud* は複数個存在するが，その中から適切な 1 台を選択する作業は *pcsd* の重要な役割である．*wncud* の選択に当たっては，『リクエスト』中にコメントの形で含まれるリクエスト発行者の login 名，所属組織，発信時刻，ページャの電話番号，各々の *wncud* の所在地などの情報をもとに，主に通信コストに着目して優先順位を決め優先度の高い (通信コストの低い) *wncud* から順番に接続を試みるという方法をとっている．

また，フロントエンドサブシステムから *pcsd* に送られてくる『リクエスト』は，抽象化された命令群で構成されているので，これらの命令群を NCU の機種を考慮して適切なコマンド群に変換する作業も *pcsd* が行なう．

このように，*pcsd* はフロントエンドサブシステムと *wncud* との仲介役を果たしており，*wncud* から見た場合，`pcsd` は，クライアントの一つと見なされる．

3.4　フロントエンドサブシステム

フロントエンドサブシステムは，ユーザの要求に応じた『リクエスト』を *pcsd* に送る役割を担う．*pcsd* に『リクエスト』を伝達するには，

[3] 稼働中は NCU を接続した tty ポートを占有する．このため login や uucp するためのポートとの共存は現在のところできない

1. telnet を用いて *pcsd* と接続し，手動でリクエストを入力する．
2. *pcsd* と交信して *pcsd* に『リクエスト』を送出するモジュールを，ユーザが C,perl などの言語を用いて独自に作成する．

といった方法がある．このうち 1 は，デバッグ時には有用な手法であるが実用性には乏しい．一方 2 は，ユーザインタフェースをユーザが自由に作成できるという利点がある．しかし，インタフェースを公開しても使いやすいアプリケーションをあらかじめ一つは実装しておかなければ，多くの利用者を獲得することができない．そこで上記の 2 の範疇に入るサンプルプログラムとして，

- ユーザの要求を電子メールで受け付けるインタフェース

を用意した．以下この『メールインタフェース』について述べる．

3.5 メールインタフェース

メールインタフェースは，図 3.3 のような書式の電子メールを受けとり，これを図 3.4 のような書式の『リクエスト』ファイルに変換し，*pcsd* に送り出す機能を持つフィルタである．このフィルタを，適当な疑似ユーザ名で /etc/aliases に登録しておけば，この疑似ユーザに送られたメールは全てこのフィルタに送られ，目的を達することができる．なお,『リクエスト』ファイルにはページャの電話番号やページャの型式などを記述しなければならない．ページャの電話番号に加えページャの型式が必要なのは，ページャの型式が異なれば表示する文字列が同じでも送出する DTMF 信号列が異なるためである．これらの情報は，メールの `Subject:` フィールドにあらかじめ記述しておく．

```
To: wide-pcs@is.titech.ac.jp
Date: Tue, 27 Aug 91 04:30:01 JST
Subject: 03-1234-5678 ntti

WIDE/PCS 定時報告:27 日 04 時 30 分
GW 系異常なし
～大野
```

図 3.3: メールによる表示の依頼

図 3.3の例のようにページャタイプとして「自由文型」が指定された場合には，メール本文中の連続する空白文字は 1 文字に圧縮され，その後先頭の 36 文字 (「自由文型」の上限) だけが *pcsd* に送られる．また,「カタカナ型」の場合には，英数字，記号，ひらがな（カタカナに変換される），カタカナがメールの本文に利用でき,「数字型」の場合には数字と少数の記号のみが利用できる．いずれの場合も，メールシステムが転送を保証している文字コードであれば，いわゆる「半角」「全角」のいずれを用いても影響はない．

ところで，ある文字を表示するのにどのような DTMF 信号列を送出する必要があるかはページャごとに異なるので，この違いを吸収するための処理をどこかで行なう必要となる．評価システムでは，メールインターフェース内にこの処理を組み込んだが，従来とは異る制御方式を用いたページャが出現した場合や，送出する DTMF 信号列を直接指定したいという要求に備え，0〜9, ＊, ＃,A,B,C,D の 16 種類の文字のみ受け付ける ASIS モード（透過モード）も用意されている．ASIS モードでは，メールインタフェース内部での変換処理は行われず，メール本文中の文字をそのまま送出すべき DTMF 信号とみなす．すなわちメール本文に「１２３＃」とあれば，[1] [2] [3] [#]を送出する．ASIS モードの応用については，5.2 節で述べる．

3.6　リクエストスプーラ

メールインタフェースには同時に複数のメールが送られる場合があるので，メールインタフェースと *pcsd* との間には，何らかのキューイング機構が必要となる．本来なら，必要な機能を明確にした上で新たに設計して実装すべきであるが，`WIDE/PCS` 稼働開始を優先さるために，lpr(lpd) が提供しているプリンタスプーラを流用し，リクエストを順次 *pcsd* に転送する仕組みを暫定的に用意した.（以下，この機構をリクエストスプーラと呼ぶ）．この暫定版のリクエストスプーラは,『リクエスト』を以下のように処理して *pcsd* に送っている．

1. 『リクエスト』は，lpr コマンドによってリクエストスプーラに送られる．
2. lpd は，スプールから『リクエスト』を順次取り出し，/etc/printcap の if エントリで指定されているフィルタ（以後 if フィルタ）に送る．
3. if フィルタは標準入力から読み込んだデータを必要に応じて加工する．

暫定版リクエストスプーラは，必要な機能をほぼ提供できた．加えて UNIX のプリンタ環境を流用したことが効果的に働き，lpr/lpq/lprm/lpc といったプリンタスプーラを制御するツールもそのまま活用できた．すなわち，現在どれくらいの『リクエスト』が送出を待っているかを知ったり，不要になった未送出の『リクエスト』をキャンセルする，といった機能を新たにプログラムを書くことなしに用意できた．しかし，送出失敗の際の再試行の機構だけは，lpr(lpd) の外に追加する必要があった．なぜなら，lpr(lpd) では，印刷が正常終了しなかった場合には印刷を打ち切るか直ちに再実行するかのいずれかしか選択枝がないが，プリンタではなくページャを駆動する場合には，確率は低いものの話中となる可能性があるからである．これは,「現時点でははは正常終了できないが，しばらく待って再試行すれば正常終了する可能性がある」という，プリンタを用いた印字にはない状態であるため，lpr(lpd) では対応できないのである．この場合には直ちに再試行しても呼が確立する可能性は低いので，他のリクエストを先に処理するなどして，一定時間が経過するまで処理を凍結し，その後再度実行する方が効率的である．そこで

if フィルタに，この再試行の機能を組み込むといった対策が必要となある．現在利用しているフィルタはこの機能を持たせているが，場あたり的な実装なので，第 4 章で述べるように引続き改良を行なっている．

3.7 リクエストの形式

電子メールで送られてきたユーザからのメッセージがページャに送られるまでの流れをまとめると，以下のようになる．

1. まずメールインタフェースで『リクエスト』に変換され，
2. 次にリクエストスプーラで順番を待ち，
3. その後順次 *pcsd* に送られ，
4. *pcsd* にて NCU 固有のコマンド群に変換された後，*wncud* に送りこまれる．

このように，ユーザの要求は何回かその表現を変えながら最終的にはページャの液晶ディスプレイ上に表示されるが，このうちメールインタフェースが作成する『リクエスト』の例を，図 3.4 に示す．これは，図 3.3 のメールを処理した結果である．

```
 1: %% From: hohno@is.titech.ac.jp
 2: %% Date: Tue, 27 Aug 91 04:30:01 JST
 3: %% RequestID: 2205165571.17069.0683235034
 4: %% Type: NTTI
 5: %% Now:    683235034 (Tue Aug 27  4:30:19 1991)
 6: %% Expire: 683235634 (Tue Aug 27  4:40:19 1991)
 7: CALL0312345678
 8: SLEEP 2
 9: DTMF*
10: SLEEP 6
11: DTMF*04
12: SLEEP 4
13: DTMF#0355#0341#0336#0337
14: DTMF#0131#0348#0335#0351
15: DTMF#3674#2794#4283#2580
16: DTMF#0107#0318#0323#3892
17: DTMF#0316#0320#2794#0319
18: DTMF#0316#4212#0339#0355
19: DTMF#2347#1659#3079#0442
20: DTMF#0423#0133#3471#4478
21: DTMF##
```

```
ＷＩＤＥ／ＰＣＳ
定時報告：２７日
０４時３０分ＧＷ
系異常なし～大野
```

図 3.4: リクエストの例 (左) と「自由文型」ページャ上での表示 (右)

図中，%で始まる行は，*pcsd* にとってはコメント行であるが，%%で始まる行 (1〜6 行目) にはコメントの形でページャ制御に副次的に必要となる情報が書かれている．

図 3.4 の例では 1 行目から順に,

1. リクエストを発行した利用者のアドレス
2. リクエストのもとになったメールが発送された日時
3. リクエストの識別番号
4. ページャの種類
5. リクエストを受け付けた時刻
6. リクエストが廃棄される時刻 (有効期限)

が書かれている．これらの情報は，話中のため送出できなかったリクエストを再投入するかどうかをリクエストスプーラが判断する場合や，*pcsd*が最適な *wncud* を選択する場合などに利用される．もし，これらのフィールドが全て揃っていない場合には，不完全なリクエストとみなされエラーとなる．

7 行目以降は *pcsd* に対するコマンドである．これらのコマンドは，まず *pcsd* によって逐次読みこまれ次に *wncud* が制御している NCU のコマンド体系に適合するよう変換された後，*wncud* に向けて送出される．

図 3.4 の例では，7 行目で CALL 命令を用いて指定された電話番号を呼びだし，8 から 12 行目でページャとのネゴシエーションを行なっている．具体的には，呼が確立した時に流される「こちらはポケットベルです」というアナウンスを途中で停止させ (`DTMF *`) たり，任意の漢字を送出するためのモードへの移行を指示して (`DTMF * 04`) いる．また，8,10,12 行目は，タイミングを調整するために挿入されている sleep コマンドである．このネゴシエーションの部分は，当然機種ごとに異なるが，逆に考えれば機種が決まれば一意に定まるので，本来なら 4 行めで与えられるページャ機種名をもとに自動生成するべきであるが，現在の実装ではリクエストファイル作成側が明示的に用意しなければならない．

次いで 13 から 20 行目では DTMF 命令を用いて実際に発信する DTMF 信号列を指定している．これがメッセージ本文で，この例では 32 文字あるので 160 桁の DTMF 信号列に展開されている．そして最後の 21 行目で # を 2 つ送出し (`DTMF##`), 終了の合図を送っている．

なお，7 行目の `CALL0312345678` は，`03-1234-5678` 局を呼び出す指示であるが，これはメタなコマンドであり，これをこのまま *wncud* に送ることはできない．*pcsd* はこのコマンドを受けとると，まず *wncud* に接続されている NCU の種別を確認しながら最適の *wncud* を選びだし，次に NCU の種別に応じたコマンドに変換して送出する．すなわち，NCU が PNC–3400 の場合には `CRN0312345678[CR] DCD44[CR]` というコマンドに変換し，AT モデムの場合には `ATDT0312345678[CR]`(プッシュ回線の場合) あるいは `ATDP0312345678[CR]`(ダイヤル回線の場合) というコマンドに変換してから *wncud* に送出する．また，`DTMF * 04` などの DTMF 発信の指示も，NCU の機種に応じて `PTT * 04` (PNC–3400 の場合), `ATDT * 04`; (AT モデムの場合）に変換される．

いずれにせよ，NCU の違いは *pcsd* が吸収するので，リクエストファイルは NCU の機種に依存しない形で記述できる．

3.8 運用経験と評価

上記のようなシステムで，1991 年 5 月末から試験運用を開始した．使用したページャは 2 種類で，同年 6 月末までは「数字型」を，翌 7 月からは「自由文型」を使用した．

また，サービスホストとしては，OMRON LUNA を選び同年 12 月まで使用した．その後，今日まで SONY RISC NEWS(NEWS OS 4.1R) を用いている．ワークステーションを交換したのは，筆者が利用できる計算機環境が変化したためであり，LUNA に不都合があったわけではない．また Sun ワークステーション (Sun4/280, SunOS 4.0.3) でも稼働することを確認しているが，長期間連続運用した実績はまだない．これも単に筆者の計算機環境によるもので，Sun に問題があるからではない．これ以外のワークステーションには，移植したことも稼働させたこともないが，BSD 系の UNIX であれば問題なく移植できるよう設計されている．

NCU には実験開始当初から PNC–3400 を用いているが，1991 年末からは Hayse 社の AT コマンドを理解する v.22, v22bis などの MODEM も利用できるようになった．

現在，毎日数回ずつ cron によって簡単なスクリプトを起動し，ユーザ数，ロードアベレージ，障害の有無などを収集し，ページャに表示させている．また，ネットワークの稼働状況，ディスクの消費量，特定のユーザからのメールの到着の有無などを報告するスクリプトも用意し利用している．

実験開始直後には，DTMF 信号の伝送が正しく行なわれず，その結果文字が正しく表示されないという，いわゆる「文字化け」が起こることがあった．原因はなかなか判明しなかったが，長期間にわたる運用記録の解析から，NCU(PNC–3400) の DTMF 信号の送出レベルや，送出タイミングの微妙なずれなどに問題があることがわかり解決をみた．また NCU には，マニュアルに記載のない制限がいくつかあった．一例として，DTMF 信号を送出した直後は次の DTMF 信号発信指令を受け付けない，一度に送出できる DTMF 信号には上限がある[4] ，などがある．

これらの問題点は，表面化するたびにひとつひとつ修正しなければならなかったが，地道に修正を続けた結果，現在ではきわめて順調に作動するようになり，ネットワーク管理者用メッセージ伝達システムとして十分利用できるという印象を得た．

[4] PNC–3400 の後継機種である PNC–3500 のマニュアルには，上限が 32 であることが明示されている．これは，筆者が実験的に求めて設定した上限と一致した．

第 4 章

改良

4.1 リクエストスプーラの改良

現在リクエストスプーラは，lpr(lpd) を流用しているが，既に述べたようにページャの駆動に失敗した場合の再実行のメカニズムは lpd だけではまかなえず，一定時間スリープした後に外部からリクエストを再投入している．この方式では現在スプールに入っている job は lpc/lprm/lpq によって制御できるが，一度以上送出に失敗し，再実行を控えてスリープしているリクエストに関しては，状況の把握が難しい．例えば，実行待ちのプロセスの数や それらの PID は ps コマンドで知ることができるが，既に何回再実行したかや，再実行予定時刻はいつか，などの情報を得ることは難しい．また，リクエストが再実行扱いになるとそのプロセスは順番を待ってスリープしてしまうので，プロセスが一つ増えてしまうという問題もある．

WIDE/PCS はまだ試験運用中のため，多くのリクエストが集中するような事態は生じていない．このためプロセスの数の増加はあまり問題にならないが，本格的な使用開始とともにこのことは見逃せない問題となると予想される．そこで，現在 WIDE/PCS 用の新しいスプールシステムを準備している．具体的にはスプールを 2 つ用意し，一方は lpd と同じ方式のキューとし，他方には送出に失敗したリクエストを入れておく．リクエストスプーラ管理デーモンは，1 つめのキューの中だけでなく，2 つ目のスプール内にある再実行予定のファイルも一定時間毎にチェックし，一定時間のスリープが完了したリクエストがあれば順次 1 つ目のスプール (キュー) の最後に移動させる．こうすれば，プロセス数を増加させることなしに再実行待ちの job を制御する環境を用意できる．

4.2 メールインタフェースの改良

メールでリクエストを送付する場合には，

- 宛先は，サービスホストの疑似ユーザ (例: wide-pcs@is.titech.ac.jp など) とし，

- Subject フィールドには，電話番号とページャの種別を記述

する必要があることは既に述べたが，この方法は，疑似ユーザ名，ページャの電話番号，種類といった，呼出側にはなじみのない情報をメールヘッダーに書くことを強要してお

り，加えてページャ所有者がページャを変更した場合には，その情報を呼び出し側に伝えなければサービスを受けられないという大きな問題がある．また，呼出側が電話番号だけでなく，ページャの種別まで知らなければならないというのは現実的ではない．この点を改定するために，以下のような改良を行った．

【改良 1】 まず，ページャの所有者とページャの電話番号や型式との対応については，以下のような対策をとる．

1. ページャの所有者ごとに対応表を作成し，サービスホスト上で一括管理する．
2. ページャの所有者は，随時自分の対応表を更新したり確認したりできるようにする．そのために，サービスホストと交渉する機能を付加する．具体的には，予め決められたフォーマットを用いて，サービスホスト上の管理用疑似ユーザにメールを送ることで実現する．

こうすることにより，呼出側の負担や混乱を軽減することができる．また，この方法を用いれば，ページャ所有者が複数のページャを所持しそれらを状況に応じて使い分けている場合でも対応できる[1]．

この改良により，Subject:フィールドにはページャの電話番号だけではなく，ページャ所有者のアドレスを書くこともできるようになった．

【改良 2】 次にメールアドレスを現在の ＜疑似ユーザ名@サービスホスト＞ という形式から，＜ページャ所有者名@サービスホスト＞ に変更する．これにより，利用者は，Subject: フィールドの特殊な使い方からも解放され，通常のメールと同じ感覚で利用できるようになる．この機能は，改良１でも述べた,「サービスホスト上に『ページャ所有者名』をキーにしてページャの電話番号やタイプを検索する機能」が用意されれば問題なく実装できる．

さて，サービスホストとして WIDE/PCS 専用のマシンを確保できれば実装上の問題点は特にないが，そうでない場合には通常のメールサービスと同居することになり，メールアドレスに関して混乱が起こる可能性がある．例えば，サービスホストを nsx.is.titech.ac.jp とし，このマシン上に yamada という実際のユーザが存在したとする．また，WIDE/PCS 利用登録者の中にも，yamada という名前で参照されるユーザが存在したとする．このとき，前者は nsx.is.titech.ac.jp のメールシステムが用意している「ユーザ名の名前空間」上に存在しており，後者は WIDE/PCS が提供する「WIDE/PCS 利用者の名前空間」上に存在しているから相互に何の関連もない．つまり両者は別人であり，人間宛のメールは人間に，WIDE/PCS 宛のメールは WIDE/PCS に正しく送られることが保証されなければならない．

このような場合には，サービスホスト名として例えば wide-pcs.is.titech.ac.jp といった別名を用意し，yamada@nsx.is.titech.ac.jp 宛のメールは通常のメールとして取り扱い，yamada@wide-pcs.is.titech.ac.jp 宛のメールは WIDE/PCS 宛のメールと

[1] 現時点では，1 台で全国を網羅するページャはないので，行動する地域に応じて複数のページャを使い分けることも珍しくない

して処理し，さらに，yamada@is.titech.ac.jp といったホスト名を省略したメールは人間宛 (yamada@nsx.is.titech.ac.jp と同等) とみなせば，混乱を回避できる．この振り分けは，単なる sendmail.cf の記述技術の問題であり，WIDE/PCS にメールを転送するメーラを定義し，さらにわずかな変更を行なうだけで実現することができる．

また，ホスト名，ドメイン名，sendmail.cf に細工をほどこさなくとも，WIDE/PCS ユーザのメールアドレスを *foo*-pager という形式にするなどといった工夫をすれば，通常のメーリングリストと同じく /etc/aliases の記述で解決できる．この場合，yamada 宛の通常のメールは yamada@is.titech.ac.jp に送り，yamada の所有するページャに送る場合には yamada-pager@is.titech.ac.jp にメールを送ることになる．

以上のような変更により，従来

```
宛先    To: wide-pcs@is.titech.ac.jp
題目    Subject: 03-5018-xxxx type = ntti
```

であった形式が

```
宛先    To: yamada@wide-pcs.is.titech.ac.jp
題目    Subject:    (なくてもよい)
```

あるいは，

```
宛先    To: yamada-pager@is.titech.ac.jp
題目    Subject:    (なくてもよい)
```

という形式になる．新しい形式では，疑似ユーザ名，ページャの電話番号，タイプといった情報を指定する必要がなくなり，基本的には相手の名前 (必要に応じて Subject: フィールドにオプションを書いてもよい) を指定するだけになる．すなわち，通常の電子メールとの相違点は文字数の制限だけになり，使い勝手は大きく向上する．なお現在の版では，上記の 3 つの方式はいずれも利用可能である．

第 5 章

応用

5.1 WIDE/PhoneShell, ファクシミリ との連携

WIDE/PCS は，ネットワーク管理をはじめとしてさまざまな局面で利用可能であるが，すでに触れたように WIDE/PhoneShell，FAX などと連携すればより有効に利用できる．

WIDE/PhoneShell は，DTMF 信号を用いた計算機への入力機能と，計算機出力を音声を用いて利用者へ伝達する機能を提供しているので，これを利用すれば必要な情報を DTMF 信号と音声によってやりとりすることができる．また FAX との連携は，音声ではなくハードコピーで情報を得る手段を提供し，計算機からの情報量が多い場合に有効である．

WIDE/PhoneShell の開発と FAX サービスによって，ネットワーク管理者は，計算機を直接操作できない環境にいてもさまざまな情報を得たり，計算機を操作したりすることが可能になったが，管理者が積極的に WIDE/PhoneShell を利用しない限り，情報が管理者に伝達されない点が問題として残っていた．すなわち重大なネットワーク上の障害の発生を見逃してしまうか発見が大幅に遅れてしまう危険性があった．

WIDE/PCS は，伝達すべき情報が発生したことを管理者に直ちに知らせる手段として，上記の問題の有効な解決策となる．この時もし情報量が多くて直接ディスプレイに情報を表示するのが困難であっても，以下のような選択を管理者に促すことができる．

1. 計算機が利用できる地点に移動する．
2. 他の管理者に電話で連絡をとる．
3. WIDE/PhoneShell を操作して音声で情報を得る．
4. WIDE/PhoneShell を操作して FAX 番号を指定し，FAX に情報を転送する．

したがって，WIDE/PCS を活用した典型的なシナリオとしては，

1. 障害発生．障害検知システムが WIDE/PCS にメッセージの伝達を依頼．
2. ページャに障害発生を伝えるメッセージが表示される．
3. ページャのメッセージを見た管理者は，WIDE/PhoneShell を操作して，状況を音声で確認する．
4. 詳しい情報を FAX に転送させる．

5. 対応を決め，WIDE/PhoneShell を操作をする．必要ならば他の管理者に応援を頼む．

6. 対処した結果の詳細を FAX に出力し障害からの復旧を確認する．
7. 障害再発が予想される場合には，その部分を重点的に監視し問題が発生したら直ちにページャにメッセージを表示するように障害検知システムの設定を変更する

といったものが考えられる．もし，小型の端末とモデムが利用可能な場合には，3 以降は，端末を用いて行った方が簡単だが，障害発生を即座に伝える WIDE/PCS の重要性に変わりはない．

5.2　その他の応用例

最近，DTMF 信号で ON/OFF できる電源スイッチが市販されるようになった．この装置を用いれば，電源を再投入するしか復旧方法のない状態に陥ってしまったモデムやルータなどを，遠隔操作で再起動することができる．もちろん，このような状態にならないように最善の努力を講じることが重要であるが，今後ゲートウェイ装置はますます増加し，またさまざまな場所に設置されるようになるので，障害を起こした装置の近傍に管理者がいないという事態も十分予想されることである．このとき，WIDE/PCS を用いれば DTMF 信号を制御して電源を再投入する作業を遠隔地から自動的に行なえる．こういったメカニズムが提供されることの意義は大きく，システムの安定運用に貢献すると思われる．

また，WIDE/PCS は計算機システム管理以外の分野にも応用することができる．

最近， VTR や空調装置などの家電製品のなかには，DTMF 信号を送ることによって録画や温度設定などを制御できる機種がある．この方法は，パーソナルコンピュータやワークステーションなどの設備がなくても，外部から機器を制御できるため便利な機能であるが，何桁もの DTMF 信号を誤りなく送出しなければならないので，操作性が優れているとは言い難い．WIDE/PCS は，このような目的にも利用することもできる．

なお，これらの場合，利用者はメールインタフェースの「ASIS モード」を利用するか，*pcsd* と通信する独自のクライアントを用意する必要がある．後者の場合，制御対象の機器を制御するのに必要な DTMF 信号列を，メニューやアイコンの操作を利用して会話的に生成するよう設計すれば，遠隔制御は現在の方法に比べて操作性の面で大幅に改善されるであろう．

第 6 章

おわりに

本論文では，ページャを用いたメッセージ転送システム WIDE/PCS について報告した．

ページャは表示できる文字数に限度があるため，大量の情報を伝達する手段としては十分ではないが，WIDE/PhoneShell や，FAX サービスと併用することにより，ネットワーク管理を強力に支援する機構を構成できる．また，WIDE/PhoneShell が抱えていた「利用者側が積極的に利用しない限り情報を伝達できない」という問題は，WIDE/PCS の導入により解決され，システム側が管理者に対して問題の存在を即座に伝達できるようになった．

今後は実際のネットワーク管理に本格的に導入し，長期間の運用経験をもとにより洗練された管理機構に成長させたい[1]．

[1] 1992 年 6 月に神戸で INET'92 が開催される．この会議の参加者のために設置されるネットワークの管理や，主なメンバーとの連絡用に WIDE/PCS が利用されることになっている．